

# PANASONIC REAL-TIME MEETING ROOM STT

Patrick Nguyen<sup>1</sup>

Panasonic Digital Networking Laboratory / Multimedia Group<sup>2</sup>

Panasonic Technologies Company  
3888 State Street, Suite 202,  
Santa Barbara, CA 93105, U.S.A.  
patrick.nguyen@ieee.org

## ABSTRACT

In this paper, we describe a real-time speech-to-text (STT) system for Meeting Room (MR) recognition developed at Panasonic. The system is an evolution of Panasonic's Broadcast News (BN) STT system that was evaluated at the NIST Rich Transcription (RT) 03S event. Newest features of interest include syllable models and merged Multiple Heteroscedastic Linear Discriminant Analysis (MHLDA) feature transformations. Also, we will present some experience that we have acquired in working with Meeting Room data.

## 1. INTRODUCTION

The Meeting Room (MR) domain has received a lot of attention from corporate and governmental funding lately. Products and applications are easy to imagine for a reasonably accurate technology. From the scientific point of view, it offers many new compelling challenges to solve. It is still in its infancy; departure from standard scoring and decoding architectures is being seriously considered.

At Panasonic, we have decided to evaluate the difficulty of the task from the Broadcast News point of view. This paper will present the snapshot of our BN-STT system that was adapted to MR. Firstly, we will describe two features that were added since the last NIST STT evaluation. The first was motivated by the sudden surge in amount of available data for training. We have upgraded to larger syllabic models from triphonic models. The second feature is our extension of MHLDA modelling. Secondly, we will expose our practical findings concerning MR data.

## 2. NEW ACOUSTIC MODELLING FEATURES

The two main acoustic modelling features that were introduced lately in our STT system were syllable models and merged MHLDA models.

### 2.1. Syllable models

We believe that the increase in amount of training data has propelled larger units modeling, and in particular syllable models, to the list of interesting features to try. Furthermore, it is argued that syllable units are intrinsically more robust to variations in pronunciations. This is particularly useful when dealing with spontaneous speech. One drawback is that syllable units are not language independent: applicability and performance might vary across languages. This dependence is expected to disappear when we move to word models as more data are introduced.

#### 2.1.1. Syllabification

Our initial lexicon was designed for careful speech. It contains all schwas. That minimizes the amount of ambiguous syllabification. All syllables will also contain at least one vowel. For instance, the word *little* is phoneticized as *l ih t ax l*, as subsequently syllabified as {*l ih*} {*t ax l*}. The NIST tool *tsyl* was used [1].

We introduce a new piece of information that is not fully covered by the syllable literature: as with allophones, the position of the syllable in the word is of utmost importance. In our example, the second syllable is *t ax l*, the same as in *battle* but not the middle of *natalya*. Since many words are monosyllabic, this is also approximately equivalent to word modeling. When introducing context, we take into account the neighboring phone. It was more manageable than using syllable context. Another possibility would be nucleus context.

Following the work of [2], when a syllable is not in our list, we use phoneme models. For simplicity, we use CI phone models. Context-dependent model should perform better, however, as we will see, since we have good coverage, the back-off strategy is only of negligible importance.

Our state-based syllable lexical tree is about twice the size of the state-based triphonic lexical tree. It is crucial to have a state-based architecture as opposed to a model-based lexical tree.

#### 2.1.2. More data

There has been many successful attempts at using syllable units in spontaneous and non-spontaneous speech. In the current literature (e.g. [3, 2, 4]), the most cited limit for performance was lack of training data and coverage of syllables. In 2003, we happened to have introduced the largest acoustic training database to have ever been used in a NIST evaluation [5]. It is appropriate for prototyping syllable experiments. We study the coverage of syllables in this corpus.

Unlike some other languages, in the English language there is virtually no hard limit on the number of distinct syllables. It is expected that no more than 20k syllables should be enough to cover the entire dictionary. By construction, we have only in-vocabulary items in the training words. Therefore, there is a hard limit defined by the lexicon of 64k words that was used for decoding the training set. There are about 15k such syllables in the lexicon. Almost all of those were seen at least once in the training set.

The training data included instances of about 15M syllables. If we select the top *N* syllables by frequency of occurrence in the training data, we can measure the coverage in Table 1. There are columns for context independent (CI-syl), position dependent (POS-syl), and word internal context position dependent (POS-CD-syl) syllable units. In our approach, there is little theoretical reason for choosing a low count of syllables because the system size will be reduced by entropy after-

<sup>1</sup> Unemployed; All work done at Panasonic Technologies Company.

<sup>2</sup> Formerly Panasonic Speech Technology Laboratory.

wards. However, practical constraints put a cap on the trainability of the system. There are about 3.33 phones per position dependent syllable on average. The untied 6k syllable system has about 60000 states. We found this to be reasonable and marginally inferior to the 10k system. Comparing with [2], we can now see the leverage in amount of data: there is no longer a need to be careful about hybridation, because our coverage is so large that backed off non syllabic units do not matter. This is quite unlike [2] and [4], where lack of data forces a careful tuning of hybridation. We will observe that the difference in performance between CI-syl and POS-syl is compelling. However, the POS-CD syl system is only marginally better than the POS-syl. From the table we can already see that the POS-CD syl system will be more complex w.r.t. the POS-syl, than the POS-syl is to the CI-syl.

# of syllables	CI-syl	POS-syl	POS-CD syl
10k	99.7%	99.5%	94.7%
6k	99.7%	98.5%	90.7%
3k	98.2%	95.0%	82.2%
2k	95.9%	91.1%	76.0%
1.5k	93.4%	87.6%	71.3%
1k	88.7%	81.6%	64.6%

**Table 1.** Coverage of the training data w.r.t. the most frequent syllables.

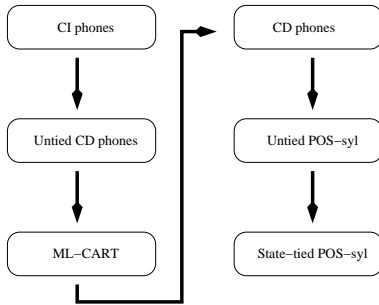
In Table 2, we show the number of examples for the  $N^{\text{th}}$  POS-syl.

$N$	# of instances
10k	1
6k	14
3k	127

**Table 2.** Number of times the  $N^{\text{th}}$  pos-syllable is seen.

### 2.1.3. Training procedure

We found out that the following training / initialization works best in practice. The training procedure is shown on Figure 1. In particular, following [3], we have also observed that seeding with CD-phone models worked best. The syllable training procedure is almost identical to the CD-phone initialization procedure off CI-phones, except that entropy merging replaces decision tree splitting. The final system has 6000 POS-syl and 3000 states. In separate experiments, we found



**Fig. 1.** Training syllables from bootstrap CD phones

a small improvement by augmenting the syllables with a phone context. The training procedure became more involved. For the sake of

simplicity and fast prototyping, we decided to use context-independent syllables only. POS-CD syllables, however, always performed best in our experiments.

## 2.2. Merged MHLDA

Semi-tied models [6], and their mathematical extension, Multiple Heteroscedastic Linear Discriminant Analysis (MHLDA) [7, 8], have been found to improve performance in the past. Here, we use MHLDA models. In some instances, especially with more data, we have found that increasing the number of transformations can also increase the accuracy. As with the number of Gaussians, there is an optimal number of transformations that most favorably balances model resolution and trainability. Unlike merging Gaussians, however, merging transformations does not have a closed form solution. We see however that there is an EM formulation that is trivially close to the STC [6] or MLLU [9] framework. We show how to use an existing MLLU implementation to estimate the best transformations. The extension to MHLDA is straightforward.

### 2.2.1. Estimating the transformation

The log likelihood of independent frames  $o_t$  with posterior  $\gamma(t)$  given a Gaussian distribution  $\mathcal{N}(\mu, C)$ :

$$Q = -\frac{1}{2} \sum_t \gamma(t) \left\{ D \log 2\pi + \log |C| + (\mu - o_t)^T C^{-1} (\mu - o_t) \right\}. \quad (1)$$

In the case of semi-tied transformations [8],  $\mathcal{N}(A\mu, ACA^T)$

$$Q = -\frac{1}{2} \sum_t \gamma(t) \left\{ D \log 2\pi + \log |C| + \log |A|^2 + (\mu - Ao_t)^T C^{-1} (\mu - Ao_t) \right\}.$$

Now imagine that  $o_t$  are created by two STC Gaussians with parameters  $\mathcal{N}(A_m \mu_m, A_m C_m A_m^T)$  with  $m = 1, 2$ , with posteriors respectively  $\gamma_m = \sum_t \gamma_m(t)$ . Matrices  $C_m$  are diagonal. The expected log-likelihood of the  $o_t$  produced by  $\mathcal{N}_m$  evaluated on themselves, is the entropy:

$$Q_m = -\frac{1}{2} \gamma_m \left[ D(1 + \log 2\pi) + \log |C_m| + \log |A_m|^2 \right]. \quad (2)$$

We would like to merge  $A_1$  and  $A_2$  into a single matrix  $A$ . The new expected likelihood of  $o_t$  produced by  $\mathcal{N}_m$  and evaluated with  $\mathcal{N}(A\tilde{\mu}_m, A\tilde{C}_m A^T)$ , are thus:

$$\tilde{Q}_m = -\frac{1}{2} \sum_t \gamma_m(t) \left[ D \log 2\pi + \log |\tilde{C}_m| + \log |A|^2 + (\tilde{\mu}_m - Ao_t)^T \tilde{C}_m^{-1} (\cdot - \cdot) \right].$$

Computing the expectation,  $\sum_t \gamma_m(t) \cdot$ , results in:

$$\tilde{Q}_m = -\frac{1}{2} \left[ D \log 2\pi + \log |\tilde{C}_m| + \log |A|^2 + (\tilde{\mu}_m - AA_m^{-1} \mu_m)^T \tilde{C}_m^{-1} (\cdot - \cdot) + \text{tr} \left\{ C_m A_m^{-T} A^T \tilde{C}_m^{-1} A A_m^{-1} \right\} \right]$$

The total likelihood is given by:

$$\tilde{Q} = \tilde{Q}_1 + \tilde{Q}_2. \quad (3)$$

There are three strategies:

1. optimize  $A$  with fixed  $\tilde{\mu}_m, \tilde{C}_m$  and iterate,
2. optimize  $A$  with fixed  $\tilde{C}_m$  and iterate, or
3. optimize  $A$  and  $\tilde{\mu}_m, \tilde{C}_m$  jointly.

Let us concentrate on the third, most general case. Differentiating wrt  $\tilde{\mu}_m$  and setting to zero is solved by:

$$\tilde{\mu}_m = AA_m^{-1}\mu_m. \quad (4)$$

Replacing back into  $Q_m$ , we have:

$$\tilde{Q}_m = -\frac{1}{2} \left[ D \log 2\pi + \log |\tilde{C}_m| + \log |A|^2 + \text{tr} \left\{ C_m A_m^{-T} A^T \tilde{C}_m^{-1} A A_m^{-1} \right\} \right]. \quad (5)$$

We can stop at that stage or proceed. Differentiating wrt to  $\tilde{C}_m$ , and setting to zero is solved by:

$$\tilde{C}_m = \text{diag} \left( A A_m^{-1} C_m A_m^{-T} A^T \right). \quad (6)$$

Define  $M_m = A_m^{-1} C_m A_m^{-T}$ , and replacing into the expression of  $Q$ , we get:

$$\tilde{Q}_m = -\frac{1}{2} \gamma_m \left[ \log |A|^2 + \log |\text{diag} AM_m A^T| \right] \quad (7)$$

up to a constant since  $\text{tr} P \text{diag}(P)^{-1} = D$ . Adding both  $Q_m$  and differentiating wrt  $A$  gives:

$$\frac{\partial \tilde{Q}}{\partial A} = -\frac{1}{2} \left[ 2(\gamma_1 + \gamma_2) A^{-T} + \gamma_1 \left\{ \text{diag}(AM_1 A^T) \right\}^{-1} M_1 A + \dots \right]. \quad (8)$$

Even row-by-row, this is difficult to optimize (see MLLT [10]).

We chose to stop at step 2, and perform a two-step optimization:

1. Initialize:  $\tilde{C}_m = C_m, A = I$
2. Estimate  $U$  for  $A \leftarrow UA$  ( $U$  is upper)
3. Estimate  $\tilde{C}_m$
4. Estimate  $L$  for  $A \leftarrow LA$ , with  $L$  lower.
5. Check for convergence and go back to step 2 if required.

For upper triangular matrices  $(A)_{kj} = a_{kj}$ , we have  $a_{kj} = 0$  if  $k > j$ . We go row by row. Let us fix a row  $d$  and drop the  $d$  index for convenience. Define  $a = [a_{d,d+1}, a_{d,d+2}, \dots, a_{dN}]$ . We have:

$$\frac{\partial \tilde{Q}}{\partial a} = -[(G_1 + G_2)a + a_{dd}z], \quad (9)$$

and thus:

$$a = -a_{dd}(G_1 + G_2)^{-1}z, \quad (10)$$

with  $G_m$  and  $z$  appropriately defined as:

$$(G_m)_{kj} := \gamma_m r_d^{(m)} m_{kj}^{(m)} \quad (11)$$

$$z_j := \gamma_1 r_d^{(1)} m_{dj}^{(1)} + \dots \quad (12)$$

For the optimization of  $a_{dd}$ ,

$$\frac{\partial \tilde{Q}}{\partial a_{dd}} = - \left[ \frac{\gamma_1 + \gamma_2}{a_{dd}} - f^T a + \varphi a_{dd} \right] \quad (13)$$

with

$$f_j := \gamma_1 r_d^{(1)} m_{jd} + \dots \quad (14)$$

$$\varphi := \gamma_1 r_d^{(1)} m_{dd} + \dots \quad (15)$$

and therefore  $f = z$ . Continuing the derivation by replacing  $a$  with its value

$$\frac{\partial \tilde{Q}}{\partial a_{dd}} = -\frac{1}{a_{dd}} \left[ \gamma_1 + \gamma_2 + \left( \varphi - f^T (G_1 + G_2)^{-1} z \right) a_{dd}^2 \right], \quad (16)$$

and thus:

$$a_{dd} = \sqrt{\frac{\gamma_1 + \gamma_2}{\varphi - f^T (G_1 + G_2)^{-1} z}} \quad (17)$$

### 2.2.2. Implementation using MLLU

Alternatively it is more convenient to use an existing MLLU implementation, which would collect, per class, and per dimension,

$$\gamma = \sum_m \gamma_m \in \mathbb{R} \text{ (one for all dimensions),}$$

$$\varphi = \sum_m \gamma_m r_d^{(m)} m_{dd} \in \mathbb{R},$$

$$z = \sum_m \gamma_m r_d^{(m)} m_{dj} \in \mathbb{R}^{N-d},$$

$$G = \sum_m G_m \in \mathbb{R}^{(N-d) \times (N-d)}.$$

In the original MLLU paper [9], this means:

$$z' := 0,$$

$$y' := z,$$

$$M' := G,$$

$$\alpha' = \varphi - z^T G z,$$

$$\beta' = 0,$$

$$\eta = \alpha^{-1} \gamma$$

As usual, the combination of those sufficient statistics is done through canonical addition. We have the memory since it is never more than what was used during MLLU training.

### 2.2.3. Evaluating the likelihood

We have found a solution for estimating the locally optimal solution to merge two transformations. Over all possible merges, we have to find the one that results in the least loss in likelihood. It is not realistic to store all of the optimal merge matrices for each pair. Therefore, the evaluation of the likelihood must be performed quickly. The evaluation of the likelihood is given by (eq. 5). If we go half a step further in EM and update the covariance, then we can evaluate with (eq. 7). This is inconvenient, however, since  $M_m$  matrices are required for every Gaussian. To simplify, we use the Hadamard inequality, which allows us to compare diagonal covariance  $\tilde{C}_m$  with full-covariance  $\tilde{C}_m$  as the upper bound,

$$\begin{aligned} \tilde{Q} &= -\frac{1}{2} \sum_m \gamma_m \left[ \log |A|^2 + \log |\text{diag} AM_m A^T| \right] \\ &\geq -\frac{1}{2} \sum_m \gamma_m \left[ \log |A|^2 + \log |AM_m A^T| \right] \\ &= -\frac{1}{2} \sum_m \gamma_m \left[ 2 \log |A|^2 + \log |M_m| \right] \\ &= -\frac{1}{2} \sum_m \gamma_m \left[ 2 \log |A|^2 - \log |A_m|^2 + \log |C_m| \right], \end{aligned}$$

which we should compare against the original untied likelihood:

$$Q = -\frac{1}{2} \sum_m \gamma_m \left[ \log |A_m|^2 + \log |C_m| \right]. \quad (18)$$

We therefore seek to maximize an upper bound for the likelihood change. The EM algorithm gives us a lower bound. The likelihood change is bounded by

$$\Delta Q \geq -\sum_m \gamma_m \log \frac{|A|^2}{|A_m|^2}, \quad (19)$$

which is strikingly similar to the Gaussian merging (Fisher) ratios. In practice we use this bound and a single iteration of upper triangular MLLU. We found no significant difference with the full method with exact likelihood computation and fully iterated MLLU matrices.

#### 2.2.4. Algorithm

To summarize, our merging algorithm is as follows:

1. For all transformation, re-create MLLU accumulators by either running a Baum-Welch or using Section 2.2.2.
2. For all classes  $c$ , compute  $\sum_m \gamma_m |A_c|$  where  $m$  is a Gaussian assigned to class  $c$ .
3. For all pairs of matrices, evaluate all distance pairs by:
  - (a) Adding both MLLU accumulators obtained in step 1 by direct sum,
  - (b) Running the MLLU solver to obtain the putative merge matrix  $A$  (only one iteration),
  - (c) Evaluating the merge likelihood loss using the approximate version of equation (19).
4. Take the pair with lowest associated likelihood loss and perform the merge. Recompute matrix and distances w.r.t. all other matrices.
5. Repeat last step until a number of transformations is met.
6. Compute merged transformations and new means.

### 3. MEETING ROOM DOMAIN

The Meeting Room domain offers many interesting problems. Speech found in meeting room is spontaneous. It is much more natural and directed than Switchboard speech. Moreover, the most compelling condition involves recognizing speech from distant microphones. This has long been considered an enormous challenge for speech recognition. On the other hand, MR data is high bandwidth (16kHz or more) with multiple microphones.

#### 3.1. Multiple microphones

##### 3.1.1. Microphone combination

The evaluation plan states that the primary condition is the multiple distant microphone condition. Also, the real-time factor is counted with respect to the true duration of speech, that is, regardless of how many input microphones are present. For simplicity, and to avoid increasing the real-time factor, we have elected to use a single stream architecture. Streams are combined at the feature level, time-synchronously, before any further processing. The combination is shown on the following figure.

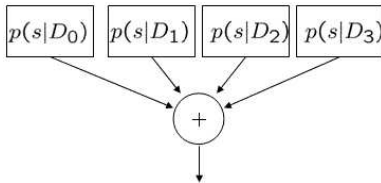


Fig. 2. Combination of features by speech posterior

The features are combined linearly. We run a GMM scorer over all channels in parallel. If there are  $C$  channels and assuming that

there is speech, each channel  $c$  has a speech log likelihood score of  $\log p(o_c^t|s)$ , where  $o_c^t$  is the channel's  $t^{\text{th}}$  frame. We choose to “flatten” the likelihood scores (e.g. [11]) and assign a channel weight  $p(s|c)$  as:

$$p(s|c) = \exp \left[ \left( \log p(o_c^t) - \log \sum_{k=1}^C p(o_k^t|s) \right) \eta \right], \quad (20)$$

with  $\eta$  chosen to be arbitrarily 0.1.

This is a simple feature combination scheme which generalizes to a Bayesian Network or an HLDA model. It was found to improve accuracy between 0% WER to 2% WER versus choosing an arbitrary distant microphone.

##### 3.1.2. MDM vs IHM

The evaluation plan states that the primary condition should be Multiple Distant Microphones (MDM). In addition, it was required as a contrast to run the same system on Individual Headmounted Microphones (IHM). The objective is to measure the difference in difficulty when distant microphones are used. The scoring, however, used a speaker-attributed (SA-STT) philosophy that requires all words to be channel-labelled, and counts errors when words are assigned to the wrong channel. Therefore, a mis-assignment counts as an insertion and a deletion. Since by design our system was geared towards the MDM condition, after stream combination, it does not know what is the source. To comply to the scoring, we added a multiplexer that selected the most likely channel for each word in a 1 sec window about its center. This was never tested experimentally. Combined with a small  $\eta$  flattening value, and the MDM-only training, might explain the exceedingly large error measured by NIST.

##### 3.1.3. Multiple speakers

Since it is not part of the scoring, we have decided to ignore effects of overlapping speech. The RT03 NIST scorer will remove entire utterances with overlapping speech. This is rather conservative. Therefore, there should not be any language model memory carried over to non-overlapping speech. This is a serious issue that will have to be addressed in the future.

#### 3.2. Bootstrapping

Except for the high bandwidth, the Meeting Room does not have much in common with Broadcast News speech. We did, however, achieve to adapt to the MR data to a certain degree. We have found that our BN system did show an 80% WER on MR without any tuning. This can be reduced to a 72% WER with LM interpolation (using BN, SWB, and MR), and vocabulary tuning. Further, with acoustic adaptation, segmentation re-training, and some tuning, we obtain an approximate 60% WER.

#### 3.3. MR statistics

We list a few statistics collected on the MR training data. The BN-64k OOV is the out-of-vocabulary rate with respect to our Broadcast News 64k vocabulary. They are shown on Table 3.

### 4. CONCLUSION

In this paper, we have attempted to describe our Meeting Room recognizer. It is based on our Broadcast News real-time decoder. We have described our newest features: firstly, a syllable unit, and secondly, merged MHLDA transformation. We have also described our adaptation of the recognizer to meeting room condition.

Stat / Corpus	ICSI	ISL	NIST
# meetings	75	19	19
# hours	72h	11h	13h
# words	622k	122k	121k
# distant mikes (avg)	4.6	1 (mix lapel)	2.8
# speakers	62	31	48
# Turns	83k	11.6k	14.8k
Turn length (words)	7.5	10.5	8.2
Turn duration	2.4s	3.2s	3.9s
BN-64k OOV	4.8%	3.4%	1.2%

**Table 3.** Statistics about the MR training data

Our one-pass real-time decoder scored at 60.58% WER in the MDM condition and 146% WER in the IHM condition.

## 5. REFERENCES

- [1] W. M. Fisher, “tsyl: NIST Syllabification Software,” available at <http://www.nist.gov/speech/tools>, June 1997.
- [2] A. Ganapathiraju, V. Goel, J. Picone, A. Corrada, G. Doddington, K. Kirchhoff, M. Ordowski, and B. Wheatley, “Syllable - A Promising Recognition Unit for LVCSR,” in *ASRU Santa Barbara*, Dec. 1997.
- [3] A. Sethy and S. Narayanan, “Split-Lexicon based Hierarchical Recognition of Speech Using Syllable and Word Level Acoustic Units,” in *ICASSP*, Hong Kong, April 2003, vol. 1, pp. 772–776.
- [4] A. Sethy, B. Ramabhadran, and S. Narayanan, “Improvements in English ASR for the MALACH Project Using Syllable-Centric Models,” in *ASRU*, Dec. 2003, pp. 129–134.
- [5] P. Nguyen and J.-C. Junqua, “PSTL’s BN-STT system,” in *Rich Transcription Workshop*, Boston, MA., 2003.
- [6] M. J. F. Gales, “Semi-tied covariance matrices for hidden Markov models,” *IEEE Transactions on Speech and Audio Processing*, no. 7, pp. 272–281, 1999.
- [7] N. Kumar and A. G. Andreou, “Heteroscedastic Linear Discriminant Analysis and Reduced Rank HMMs for Improved Speech Recognition,” *Speech Communication*, no. 26, pp. 283–297, 1998.
- [8] M. J. F. Gales, “Maximum Likelihood Multiple Projection Schemes For Hidden Markov Models – TR.365,” Tech. Rep., Cambridge University (CUED), Nov. 1999.
- [9] P. Nguyen, L. Rigazio, C. Wellekens, and J.-C. Junqua, “LU Factorization for Feature Transformation,” in *ICSLP*, Boulder, USA, 2002.
- [10] R. A. Gopinath, “Maximum Likelihood Modeling with Gaussian Distributions for Classification,” in *ICASSP*, Seattle, 1998.
- [11] P. Nguyen, P. Gelin, J.-C. Junqua, and J.-T. Chien, “N-Best Based Supervised and Unsupervised Adaptation for Native and Non-Native Speakers in Cars,” in *ICASSP*, Phoenix, Arizona, May 1999, vol. 1, pp. 173–176.
- [12] M. J. F. Gales, “Maximum Likelihood Linear Transformations for HMM-based Speech Recognition – TR.291,” Tech. Rep., Cambridge University (CUED), May 1997.